I hereby certify that this is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR § 1.10 on the date indicated below and is addressed to:

Assistant Commissioner for Patents
Box Patent Application
Washington, D.C. 20231

By: Teresa A. Fleming

Typed Name: Teresa A. Fleming

Express Mail Label No.: EL 828142439 US

Date of Deposit: 7/13/01

Attorney Docket No.: SUN-P5493-MDF

METHOD FOR MERGING A PLURALITY OF DATA STREAMS INTO A SINGLE DATA STREAM

Inventor: James M. Avery

This patent application discloses subject matter that is related to the subject matter disclosed in United States Patent Application Serial No. __/____ entitled "Apparatus for Merging a Plurality Of Data Streams Into a Single Data Stream" filed on even date herein. The above Patent Application is hereby incorporated by reference.

1. FIELD OF THE INVENTION

The present invention generally relates to arbitration methods. More specifically, the present invention relates to a novel method of merging a plurality of data streams into a single data stream.

2. BACKGROUND

In an effort to increase I/O bandwidth in high performance processor based systems, a number of companies have developed the HyperTransport ("HT") I/O interconnect structure. Briefly, the HT I/O interconnect structure is a scalable device level architecture that provides a significant increase in transaction throughput over existing I/O bus architectures such as Peripheral Component Interconnect ("PCI") and Advanced Graphics Port ("AGP").

The foundation of the HT I/O interconnect is dual point-to-point unidirectional links consisting of a data path, control signals, and clock signals. The HT I/O interconnect can provide both point-to-point links and a scalable network topology using HT I/O switching fabrics. Thus, an HT based system can be expanded using HT switches to support multilevel, highly complex systems.

Communications between multiple HT I/O devices are known as data streams. Each data stream contains one or more packets of information. Each packet of information contains a packet ID and a data payload. The packet ID is also commonly referred to as a unit ID. Because all packets are transferred to or from a host bridge, the packet ID provides information that can be utilized to determine the source or destination of the packet. A more detailed description of the HT I/O interconnect structure is presented in Appendix A.

Figure 1 presents an HT I/O device 100 that interfaces with a first unidirectional link 110 and a second unidirectional link 120. Thus, the HT I/O device 100 can receive input data streams and transmit output data streams via unidirectional links 110 and 120. The HT I/O device 100 contains input ports 130 and 150 for receiving data streams and

output ports 140 and 160 for transmitting data streams. The HT device 100 may also contain circuitry for generating packets that can be transmitted as output data streams via the output ports 140 and 160.

HT I/O devices may also be daisy chained as shown in Figure 2. Figure 2 presents a portion of a single unidirectional link in an HT I/O interconnect. The unidirectional link shown contains three HT I/O devices 210, 220, and 230. If the first HT I/O device 210 receives a data stream with a destination ID that is equal to the ID of the first HT I/O device 210, then the first HT I/O device 210 will receive and internally process the data stream. However, if the destination ID is not equal to the ID of the first HT I/O device 210, then the first HT I/O device 210 will forward the data stream to the second HT I/O device 220.

As the first HT I/O device 210 may also have the capability to generate packets, the output data stream of the first HT I/O device 210 is a composite of the input packet stream received by the first HT I/O device 210 and the internally generated packets. These internally generated packets will be referred to as an internal data stream.

The data stream received by the first HT I/O device 210 and the device's internal data stream may vary with time. For example, the input data stream for the first HT I/O device 210 may contain no packets over a given time interval. Thus, all packets in the internal data stream generated during that time interval by the first HT I/O device 210 may be transmitted through the first HT I/O device's output port. Alternately, if the data stream received by the first HT I/O device 210 and the device's internal data stream both contain a large number of packets, the HT I/O device may be required to choose between forwarding the received data stream or outputting the internally generated packets. The

process by which such a choice is made is known in the art as a forwarding fairness algorithm.

Prior art systems allow an HT I/O device to insert internally generated packets into an output data stream freely if the output data stream is empty. However, if the output data stream contains a large number of packets, the prior art systems only allow the HT I/O device to insert internally generated packets into the output data stream at a rate that is not greater than the rate that the HT I/O device is receiving and forwarding packets from another HT I/O device. Such prior art systems are not optimal. Thus, a more optimal method of merging two data streams into a single data stream is needed.

3. SUMMARY OF INVENTION

One embodiment of the invention is a method of merging a first data stream with a second data stream to generate a third data stream. The method comprises receiving a first packet from the first data stream, the first packet containing a first packet ID and a first data payload and receiving a second packet from the second data stream, the second packet containing a second packet ID and a second data payload. The method also includes storing first data in a plurality of packet ID arrival registers, a first portion of the first data indicating that the first packet ID is equal to the ID associated with a first of the plurality of the packet ID arrival registers, a second portion of the first data indicating that the first packet ID is not equal to the ID associated with a second of the plurality of the packet ID arrival registers; storing second data in the plurality of packet ID arrival registers, a first portion of the second data indicating that the second packet ID is equal to the ID associated with the second of the plurality of the packet ID arrival registers, a

second portion of the second data indicating that the second packet ID is not equal to the ID associated with the first of the plurality of the packet ID arrival registers. The method further includes calculating a first autocorrelation vector; calculating a second autocorrelation vector; and based at least in part upon a comparison of the magnitude of the first autocorrelation vector and the magnitude of the second autocorrelation vector, including the first packet in the third data stream.

Another embodiment of the invention includes receiving the first packet from an HT I/O device.

Another embodiment of the invention includes receiving the first packet from an HT I/O device and receiving the second packet from an HT I/O device.

Another embodiment of the invention includes storing a "1" in the first packet ID arrival register.

Another embodiment of the invention includes storing a "0" in the second packet ID arrival register.

Another embodiment of the invention includes calculating a biased autocorrelation vector.

Another embodiment of the invention includes calculating an unbiased autocorrelation vector.

Another embodiment of the invention is wherein the act of calculating the first autocorrelation vector is performed by a HyperTransport I/O device.

Another embodiment of the invention is wherein the act of calculating the first autocorrelation vector is performed by a HyperTransport I/O switch.

Another embodiment of the invention is wherein the act of calculating the first autocorrelation vector includes calculating the following equation:

$$Rxx(T) = \frac{1}{N-T}\sum_{n=0}^{N-1} x(n)x(n+T)$$

where T and N are integers, and x is an array that includes data stored in one of the plurality of packet ID arrival registers.

Another embodiment of the invention is wherein the act of receiving the first packet includes receiving the first packet from a second HT I/O device and wherein the act of receiving the second packet includes receiving the second packet from a third HT I/O device.

Another embodiment of the invention is wherein the act of receiving the first packet includes receiving the first packet from an internal port within the HT I/O device and wherein the act of receiving the second packet includes receiving the second packet from a second HT I/O device.

Another embodiment of the invention is wherein the act of calculating the first autocorrelation vector includes copying the data in the first packet ID arrival register, and shifting the copied data by T elements, where T is an integer.

Another embodiment of the invention is a method, performed by an HT I/O device, of storing a first packet in a first buffer and storing a second packet in a second buffer. The method includes receiving a first packet from a first data stream, the first packet containing a first packet ID and a first data payload and receiving a second packet from the second data stream, the second packet containing a second packet ID and a second data payload. The method also includes storing the first packet in a first buffer, the first buffer associated with a buffer ID that is equal to the first packet ID and storing

the second packet in a second buffer, the second buffer associated with a buffer ID that is equal to the second packet ID.

Another embodiment of the invention is a method of storing first data and second data. The method includes receiving a first packet from a first data stream, the first packet containing a first packet ID and a first data payload and receiving a second packet from a second data stream, the second packet containing a second packet ID and a second data payload. The method also includes storing first data in a plurality of registers, a first portion of the first data indicating that the first packet ID is equal to the ID associated with a first of the plurality of registers, a second portion of the first data indicating that the first packet ID is not equal to the ID associated with a second of the plurality of the registers and storing second data in the plurality of registers, a first portion of the second data indicating that the second packet ID is equal to the ID associated with the second of the plurality of the registers, a second portion of the second data indicating that the second packet ID is not equal to the ID associated with the first of the plurality of the registers.

4. BRIEF DESCRIPTION OF THE FIGURES

Figure 1 presents an HT I/O device that interfaces with two unidirectional links.

Figure 2 presents a portion of a single unidirectional link in an HT I/O interconnect.

Figure 3 presents a portion of an HT I/O device.

Figure 4 presents one embodiment of an input port.

Figure 5 presents one embodiment of an output port.

Figure 6 presents one method of utilizing data contained in a packet ID arrival register in an autocorrelation function.

Figure 7 presents one embodiment of an autocorrelation magnitude table and one embodiment of a maximum autocorrelation magnitude table.

Figure 8 presents a portion of an HT I/O switch.

Figure 9 presents a portion of an HT I/O device with two internal ports.

Figure 10 presents a portion of an HT I/O device that sends flow control information to packet transmitters.

Figure 11 presents a flow chart of a method to store data in a plurality of registers.

Figure 12 presents a flow chart of a method to merge two data streams to generate a third data stream.

5. DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

Figure 3 presents a portion of an HT I/O device 300. The HT I/O device 300 is coupled to a unidirectional link in an HT I/O interconnect structure. The HT I/O device

300 would also typically be coupled to another unidirectional link. However, this second unidirectional link is not shown in order not to obscure the invention.

5.1 Input Port

The HT I/O device 300 can receive a data stream via the device's input port 310. The input port 310 may be any type of port that is operable to receive a data stream. In one embodiment, the input port 310 includes a plurality of input receivers configured as shown in Figure 4. Each of the plurality of input receivers can receive a single differential signal. Examples of such differential signals could include command, address, data, clock, and control signals. By including sufficient input receivers configured as shown in Figure 4, the HT I/O device 300 can receive a data stream from an HT I/O interconnect.

5.2 Internal Port

In addition to the input port 310, the HT I/O device may include an internal port 320. The internal port 320 may be any type of port that is operable to receive and optionally temporarily store, one or more packets in a data stream that were generated by the HT I/O device 300. For example, the internal port 320 may be a buffer such as a circular buffer, a first-in-first-out buffer, or a queue buffer. Alternatively, the internal port 320 may be one or more registers.

5.3 Input Buffers

As shown in Figure 3, the input port 310 is coupled to a plurality of input buffers 330. Input buffers 331, 332, and 333 may be any type of buffer, such as but not limited to circular buffers, first-in-first-out buffers, or queue buffers. Alternatively, such buffers may be a plurality of registers. Each of the input buffers 331, 332, and 333 is associated with a unique ID. As shown in Figure 3, the first input buffer 331 is associated with ID 0, the second input buffer 332 is associated with ID 1, and the third input buffer 333 is associated with ID m. These input buffers 331, 332, and 333 store packets having a packet ID that is equal to the ID associated with the input buffer.

For example, ID 0 may be set to 10h. Thus, if two packets are received by input port 310 and the packet ID for both of the packets is equal to 10h, then the packets would be stored in the first input buffer 331. Because the ID of each input buffer is unique, the packets would only be stored in a single input buffer.

5.4 Internal Buffers

In embodiments of the invention that include an internal port 320, the internal port 320 may be coupled to a single internal buffer (not shown) or a plurality of internal buffers 340. Internal buffers 341 and 342 may be any type of buffer or register. Each of the internal buffers 341 and 342 is associated with a unique ID. As shown in Figure 3, the first internal buffer 341 is associated with ID p and the second input buffer 342 is associated with ID q. These internal buffers 341 and 342 store packets generated

internally by the HT I/O device having a packet ID that is equal to the ID associated with the internal buffer.

A summary of the previously discussed method of storing packets in input buffers 330 and/or in internal buffers 340 is summarized in Figure 10.

5.5 Packet ID Arrival Registers

As shown in Figure 3, the input port 310 is also coupled to a plurality of packet ID arrival registers 350. Each of these registers is associated with a unique ID. As shown in Figure 3, the first packet ID arrival register 351 is associated with ID 0, the second packet ID arrival register 352 is associated with ID 1, and the third packet ID arrival register 353 is associated with ID m. These packet ID arrival registers 351, 352, and 353 store data that indicates whether a packet received by the input port 310 contains a packet ID that is equal to the ID associated with the packet ID arrival register.

For example, as discussed above, ID 0 may be set to 10h. Thus, if a packet is received by input port 310 and the packet ID is equal to 10h, then data, such as a "1", would be stored in the first packet ID arrival register 351. Because the ID of each packet ID arrival register is unique, data, such as a "0", would be stored in each of the other packet ID arrival registers.

A packet ID arrival register may be any type of register. However, in some embodiments of the invention, the packet ID arrival registers would be shift registers so that older data could be efficiently shifted out as new data is stored in the packet ID arrival registers.

In embodiments of the invention that include an internal port 320, the internal port 320 is also coupled to the plurality of packet ID arrival registers 350. In these embodiments, a packet ID arrival register's ID, such as the fourth packet ID arrival register's ID, ID p, shown in Figure 3, may be set to 15h. Thus, if the HT I/O device 300 internally generates a packet with a packet ID equal to 15h, and the packet is sent to the internal port, then data, such as a "1", would be stored in the fourth packet ID arrival register 354. Because the ID of each packet ID arrival register is unique, data, such as a "0", would be stored in each of the other packet ID arrival registers 351, 352, 353 and 355.

5.6 Autocorrelation logic

The packet ID arrival registers are coupled to autocorrelation logic 360. In one embodiment, for each packet ID arrival register, the autocorrelation logic 360 uses the data stored in the packet ID arrival register to calculate the autocorrelation vector, $Rxx(T)$, of the data stored in the packet ID arrival register over the interval T. The autocorrelation vector of such data can be calculated using the following equation:

$$Rxx(T) = \frac{1}{N-T}\sum_{n=0}^{N-1} x(n)x(n+T) \qquad T = (0, 1, 2, \ldots, N-1)$$

In the above equation, T and N are integers, and $x()$ is an array that includes data stored in one of the plurality of packet ID arrival registers 350. Conceptually, the summed portion of the above equation corresponds to taking data in a packet ID arrival register, shifting it by T elements, multiplying the result element by element with the unshifted packet ID arrival register, and then summing the products. Thus, the magnitude of the autocorrelation vector, $Rxx(T)$, represents the approximate arrival rate

of incoming packets. Similarly, peaks in the autocorrelation vector, $Rxx(T)$, approximate the average arrival frequency of incoming packets.

Figure 6 indicates how the data contained in the first packet ID arrival register 351 can be utilized in the above equation. The right most element in the first packet ID arrival register 351 is addressed as $x(0)$. Similarly, the leftmost element in the first packet ID arrival register 351 is addressed as $x(5)$. As discussed above, a "1" in the array indicates the arrival of a packet that contains a packet ID that is equal to the ID associated with a specific packet ID arrival register. Similarly, a "0" in the array indicates the arrival of a packet that contains a packet ID that is not equal to the ID of a specific packet ID arrival register. By selecting such values for the array, *i.e.*, "1" and "0", the multiplication product $x(n)x(n + T)$ may be replaced with $x(n)$ & $x(n + T)$. Such a replacement will reduce the complexity and die size of the autocorrelation logic 360.

In other embodiments of the invention, the above autocorrelation vector, $Rxx(T)$, is scaled. For example, the biased estimate of the autocorrelation vector:

$$Rxx_{biased}(T) = \frac{Rxx(T)}{N-1}$$

may be calculated by the autocorrelation logic 360. Alternatively, the unbiased estimate of the autocorrelation vector:

$$Rxx_{unbiased}(T) = \frac{Rxx(T)}{N-1-|T|}$$

may be calculated by the autocorrelation logic 360.

5.7 Arbiter

The packet arrival determination logic 360 is coupled to an arbiter 370 as shown in Figure 3. The arbiter 370 receives the autocorrelation vectors from the packet arrival determination logic 360 and determines which packet should be output by the output port 329.

In one embodiment, the arbiter 370 contains an autocorrelation magnitude table 700. As shown in Figure 7, each row of the autocorrelation magnitude table 700 is associated with a packet ID arrival register. In addition, each column in the autocorrelation magnitude table 700 is associated with a packet whose arrival data is stored in a packet ID arrival register. The autocorrelation magnitude table 700 may be a buffer such as a circular buffer, a first-in-first-out buffer, or a queue buffer. Alternatively, the autocorrelation magnitude table 700 may be composed of registers such as shift registers.

In addition to the autocorrelation magnitude table 700, the arbiter may also contain a maximum autocorrelation magnitude table 710. Each element of the maximum autocorrelation magnitude table 710 is associated with a row of the autocorrelation magnitude table 700, and hence, a packet ID arrival register and a packet ID. The maximum autocorrelation magnitude table 710 may be composed of any of the above buffers or registers.

After the arbiter 370 receives the autocorrelation vector, $Rxx(T)$, for each of the plurality of packet ID arrival registers 350, the arbiter 370 calculates the magnitude of

each of the autocorrelation vectors and stores the magnitudes in the autocorrelation magnitude table 700.

Next, for each row in the maximum autocorrelation magnitude table 710, the arbiter 370 calculates the maximum of the autocorrelation magnitudes in each row of the autocorrelation magnitude table 700 and places such maximum values in the maximum autocorrelation magnitude table 710. For example, if the maximum autocorrelation magnitude of the 6 elements in the autocorrelation magnitude table row associated with ID 0, as shown in Figure 7, is 10h then the value of 10h would be stored in the first row of the maximum autocorrelation magnitude table 710.

Next, the arbiter 370 determines which element in the maximum autocorrelation magnitude table 710 contains the largest autocorrelation magnitude. The arbiter 370 then issues a command to the packet selector 380, which is discussed in Section 5.8, to select a packet with the packet ID that is associated with such element.

In still other embodiments of the invention, the arbiter 370 receives $Rxx_{biased}(T)$ or $Rxx_{unbiased}(T)$ vectors instead of the $Rxx(T)$ vectors discussed above. In these embodiments, the arbitration methods would be substantially identical to the methods discussed above.

5.8 Packet Selector

The input buffers 330, the internal buffers 340, and the arbiter 370 are each coupled to a packet selector 380. The packet selector 380 can receive packets from any of the plurality of input buffers 330 or from any of the plurality of internal buffers 340. However, the packet selector 380 will receive a packet from these buffers 330 and 340

when it is commanded to do so by the arbiter 370. Thus, when the packet selector 380 receives a command from the arbiter 370 to select a packet from one of these buffers 330 and 340, the packet selector 380 receives a packet and passes the packet to the output port 390.

5.9 Output Port

As shown in Figure 3, the packet selector 380 is coupled to an output port 390. The output port may be any type of port that is operable to generate a data stream. In one embodiment, the output port includes a plurality of output drivers configured as shown in Figure 5. Each of the plurality of output drivers can generate a single differential signal, such as but not limited to, command, address, data, clock, and control signals.

5.10 Source IDs, Destination IDs, and Packet IDs

In the above description, packet IDs, which provide information relating to the source or destination of packets, were associated with specific input buffers, internal buffers, and packet ID arrival registers. However, in other embodiments of the invention, the input buffers, internal buffers, and packet ID arrival registers could be associated with any packet information that can be utilized to identify a packet.

5.11 Data Streams

As discussed above, Figure 3 presents only a portion of an HT I/O device 300. The HT I/O device 300 is shown coupled to a single unidirectional link in an HT I/O interconnect structure. The HT I/O device 300 would also typically be coupled to

another unidirectional link. However, this second unidirectional link is not shown in order not to obscure the invention. In many of the embodiments of the invention, the HT I/O device 300 would also include input buffers, internal buffers, packet ID arrival registers, autocorrelation logic, an arbiter, and a packet selector. These components would operate as discussed above to merge an input data stream from the second unidirectional link with the HT I/O device's internal data stream into a single output data stream on the second unidirectional link.

While some of the embodiments discussed above merge an input data stream and an internal data stream into a single output data stream, the invention is not so limited. Some embodiments of the invention merge a data stream from one input port with other data stream(s) from one or more input ports. For example, Figure 8 presents an HT I/O switch capable of merging data streams received from a plurality of input ports into a single output data stream.

Other embodiments of the invention, such as shown in Figure 9, would merge a plurality of internal data streams into a single output data stream. Still other embodiments of the invention would merge one or more input data streams with one or more internal data streams.

5.12 Buffer Credits

Some bus architectures, such as the HT I/O interconnect architecture, are flow controlled using a coupon-based scheme. In such bus architectures, a packet transmitter contains a counter that corresponds to the free space available in a buffer at the packet receiver, such as an HT I/O device. After initialization, the packet receiver sends packets

to the packet transmitter to indicate the free space available in the packet transmitter buffer. This information is stored in a counter in the packet transmitter. Thereafter, when the packet transmitter sends a packet to the packet receiver, the packet transmitter decrements the counter. If the counter ever reaches zero, the packet transmitter ceases sending packets to the packet receiver.

In some embodiments of the invention, such as shown in Figure 10, when the HT device 900 passes a packet to the first output port 990, the HT device also sends a packet containing flow control information to the packet transmitter via the second output port 995.

5.13 Conclusion

A summary of the previously discussed method of merging a first data stream with a second data stream to generate a third data stream is summarized in Figure 12.

The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.